LabSat 3 WB Remote Control

Reference Manual

API for programming in .NET and COM compatible languages

This page is intentionally left blank

Contents

Introduction

An API is available to allow functions of a LabSat3 WB connected via LAN to be accessed programmatically.

This document covers the basic principles of the API and information about the API methods. There is also sample code that demonstrates API usage.

Installing the LabSat 3 WB API

The Microsoft .NET framework 4.7.1 must be installed before you can use the API.

.NET Projects

In order to access the LabSat3 WB API via your .NET project, first copy the following supplied files into your application directory:

- Racelogic.Telnet.dll
- LabSatAPI.dll
- LabSat3WBAPI.dll
- NLog.dll

Make a reference to NLog.dll, LabSatAPI.dll and LabSat3WBAPI.dll in the usual way. There is no need to reference Racelogic.TelNet.dll but ensure that it is included in your project build directory.

Other COM Compatible Languages

From Windows, the LabSat3 WB API also supports integration from COM compatible languages such as Python, VBScript, C++ and VB6. To allow this, the Register.bat script in the LabSat3WBAPI folder needs to be run. Note that the API Will need to be on a local drive as opposed to a network drive. On Windows 7 and above, this will need to be run as an administrator (right click and choose Run as Administrator from the context menu). Alternatively, you can run the RegAsm command manually from a command prompt as described in the Microsoft .NET documentation.

Python

Please ensure that you are using the 32-bit version of the Python interpreter.

To access the LabSat API via Python, first follow the instructions for 'Other COM Compatible Languages' above.
Then, install the COMTypes package via your package manager – for example:

```
\pythonProject32\Scripts>pip install comtypes
```

See the included Python sample code for example of API usage in conjunction with the comtypes package.

API Commands

The API consists of a number of 'Command' classes. Each command class provides access to one or more 'Parameters' classes that contain parameters available for the command. Each parameter supplied will instruct the LabSat to perform a particular operation or change a setting.

A list of currently available commands can be found below:

- RecordCommand
- PlayCommand
- FindCommand
- AttenuationCommand
- MuteCommand
- ConfigCommand
- MediaCommand
- TypeCommand

Each command has the following key methods:

- An 'Execute' method that accepts relevant command parameters and executes the command.
- A 'QueryStatus' method that instructs the LabSat to return the status of the command, such as the configuration values or whether the LabSat is playing at the moment.

Each command also has the following key properties:

- CommandText – the Telnet command that was actually supplied to the LabSat when the command was executed. This is for information and diagnostics purposes primarily.
- ErrorEncountered – Denotes whether or not an error was encountered when executing a command.
- OutputText – Contains the output returned from the LabSat following a command.

In C#, when creating an instance of a given Command class play we must pass in the IP address of the device. This information is used to connect to the LabSat when the command is executed. For example:

```
var cmd = new PlayCommand("192.168.1.117");
```

The command can then be executed along with the parameters to required perform the operation – for example:

```
cmd.Execute(new LabSatAPI.PlayCommandParams() { FileName = filename, FromSeconds = 0.1, PlayForSeconds = 10 });
```

Intelisense will provide guidance for the parameters accepted for each command.

Example API Calls

C# sample code that demonstrates how to use the API can be found in the file entitled 'CSharpLS3WB.cs' that is provided with this document.

COM Interface

The COM interface exposed by the API allows control from non Microsoft.NET languages such as VBScript that are COM compatible. An example VBScript file is included to demonstrate API method calls via COM.

N.B. Any scripts written must be run in a 32 bit environment on 64 bit Windows. This can be achieved by calling c:\windows\syswow64\cscript.exe PathToScript.vbs

Successive API Calls

There can on occasions be issues with making multiple commands to a LabSat in succession. Should such issues be encountered it is recommended that a delay (i.e. 100ms) is introduced between API method calls to resolve.

Further API Documentation

Refer to the help file entitled 'LabSat 3 WB API Documentation.ctm' for detailed information. It is advised to copy this file locally (i.e. desktop) since security settings may prevent this file from showing all information over a network.

If you have downloaded the file via a browser, you may get a security warning whilst opening this file:
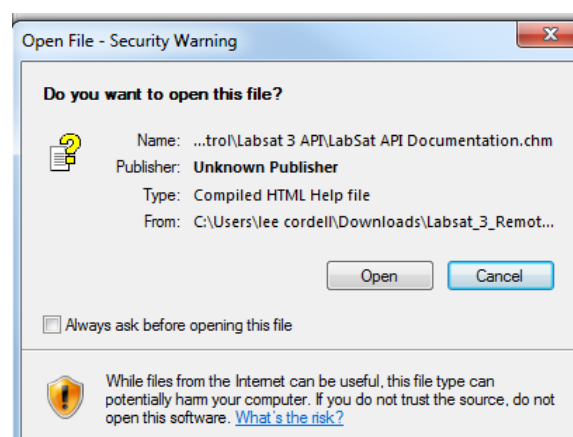
This will prevent the help file from showing all information. To resolve this issue right click on the 'LabSat 3 WB API Documentation.ctm' file that you have copied locally and click 'Properties', towards the bottom of the properties page there will be a 'Security' section stating that 'This file came from another computer…'
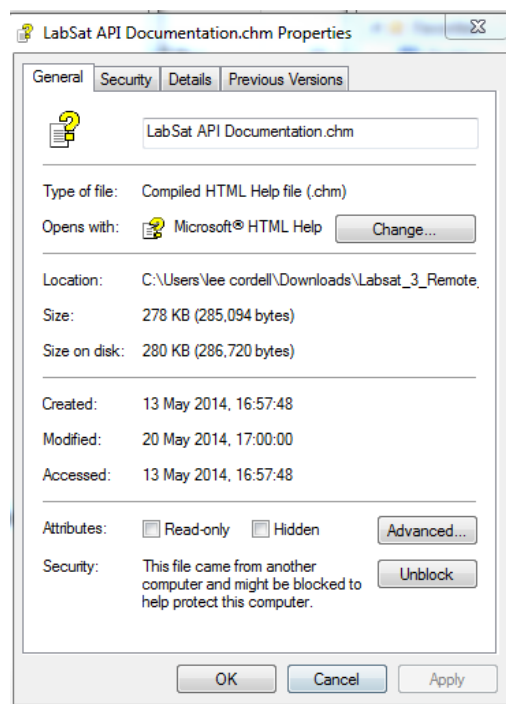
Click the 'Unblock' button to the right of this message then click 'Ok' to close the properties page. The file should now open correctly.

Contact Details

Racelogic Head Office
Unit 10, Swan Business Centre,
Osier Way Buckingham
Bucks MK18 1TB
United Kingdom

Contact: Katie Harland

Tel: +44 1280 823 803
Fax: +44 1280 823 595

Email: labsat@racelogic.co.uk

Document Version Control

Revision	Description	Date
1.0	Initial version	18/08/2017
1.1	Media command code sample added	13/12/2017
1.2	Allow multiple frequencies, bug fixes	08/03/2018
1.2.3	Record command fix for latest firmware	12/04/2018
1.2.5	RS232 Output setup added. Telnet display formatting now omitted from outputtext.	08/05/2018
1.2.6	Added Python sample code. Bug fixed with media command via COM.	30/08/2018
1.2.6_1	Added Python 3.x sample code. Tested with Python 3.10. LabSatWBAPI unchanged at 1.2.6.	23/11/2021